Agilent GeneSpring Workgroup SampleLoader 5.3

User Guide

The GeneSpring Workgroup SampleLoader automates the process of loading large volumes of microarray data directly into GeneSpring Workgroup. SampleLoader automatically updates the GeneSpring Workgroup database with data in existing LIMS and legacy databases, ensuring that data is synchronized and accessible across diverse parts of an enterprise. SampleLoader is designed to run as a standalone application on Linux and Solaris machines that is launched from an automated task scheduler (such as a Unix cron job), the command line, or another application. Samples can be loaded from virtually any SQL compliant database and from a directory containing text files by command-line options when running SampleLoader or setting options in a SampleLoader XML configuration file. In addition, SampleLoader can load samples a user-designed java class generates, allowing preprocessing of raw data before it is added to the GeneSpring Workgroup repository. SampleLoader makes it trivial to maintain a working mirror of your proprietary gene expression database within GeneSpring Workgroup.

The location of the samples and information about the upload such as the genome and the owner of the samples are specified in the SampleLoader configuration file. You can have

configuration files to cover a variety of upload situations. You can retrieve data and attributes from the same or different sources, and can specify multiple queries within the same configuration file. In addition, sample data can be loaded in batch from multiple, specified network directories, each with unique instructions for handling access permissions and identifying technology types. Associated files can also be retrieved and uploaded with SampleLoader.

Installing SampleLoader

System Requirements

Before you install SampleLoader, GeneSpring Workgroup must already be installed, configured, and running. You must also have the `SampleLoader.jar` and `SampleLoader.xml` files.

- Linux (i.e., Red Hat version 3 or above) or Solaris (version 8 or above)
- A JVM that supports JDK1.4 or later
- 500 MB RAM
- 40 GB HDD
- The appropriate JDBC driver for each external database storing sample data, if using GeneSpring Workgroup Oracle or importing samples from an external SQL database

NOTE SampleLoader should not be installed on the same system as GeneSpring Workgroup Server.

The SampleLoader distribution contains the following files:

- `SampleLoader.jar`—The SampleLoader executable
- `SampleLoader.xml`—A sample configuration file

The following seven example XML files are included in the SampleLoader directory. Pick the one that most closely matches you data loading needs and modify the necessary tags.

- `SampleLoader_MultiDB.xml`
- `SampleLoader_FlatFile_JavaClass.xml`
- `SampleLoader_FlatFileImagene.xml`
- `SampleLoader_FlatFile_DBAttributes.xml`
- `SampleLoader_FlatFileAffy.xml`
- `SampleLoader_AADM_SQLServer.xml`
- `SampleLoader_AADM_Oracle.xml`

Installation

1. Create a directory for SampleLoader, for example: `/usr/local/SampleLoader.`

   ```
   % cd /usr/local
   % mkdir SampleLoader
   ```

2. Place the following files in the new directory:
   - `SampleLoader.jar`
   - `SampleLoader.xml`
3. Create a script to execute SampleLoader. This script must specify the following information:
   - The location of the Java executable on your system
   - The directory containing the SampleLoader executable and configuration file
   - An instruction to run the SampleLoader executable using the configuration file you specify
   - An instruction to set the *DISPLAY* variable (`192.168.10.10` in the sample load script below)

   You must also add the following items to your `CLASSPATH`:
   - The JDBC driver for your external database or GeneSpring Workgroup Oracle (if necessary)
   - The location of the SampleLoader executable `-SampleLoader.jar`

The following is an example of such a script:

```
#!/bin/sh
DISPLAY=192.168.10.10:0
HOME=/usr/local/SampleLoaderJAVA_HOME=gs1_4_0jre

SAMPLE_JAR=SampleLoader.jar

export DISPLAY
export JAVA_HOME
export HOME
export SAMPLE_JAR
JAVA=$JAVA_HOME/bin/java
export JAVA

LD_LIBRARY_PATH=$JAVA_HOME/lib/rt.jar
```

```
CLASSPATH=$JAVA_HOME/lib/rt.jar
CLASSPATH=$CLASSPATH:$HOME/$SAMPLE_JAR
CLASSPATH=$CLASSPATH:$HOME/jtds-0.3.1.jar
CLASSPATH=$CLASSPATH:$HOME/classes12.zip

$JAVA -classpath $CLASSPATH SampleLoader -config
   SampleLoader.xml
```

If you want to pull data from several different databases or directories, or have multiple genomes, you can create multiple configuration files and list each one to be called in the loading script. Add another line like the last one in the above example for each configuration file. For example:

```
$JAVA -classpath $CLASSPATH SampleLoader -config
   SampleLoader_Affy_MU74.xml
$JAVA -classpath $CLASSPATH SampleLoader -config
   SampleLoader_Affy_HU133A.xml
$JAVA -classpath $CLASSPATH SampleLoader -config
   SampleLoader_GenePixResults.xml
```

Once you have created this script, you must customize the configuration file for your system before you can run SampleLoader. See “Configuring SampleLoader” on page 9 for detailed information.

Running SampleLoader

To run SampleLoader, once you have completed its configuration file, execute the script you created in the previous section. You can either execute the script manually, or add it to your cron job entries to run at specified intervals. Agilent recommends running the load script from the command line for the first upload, especially if you have a large number of samples. Once you have run SampleLoader for the first time, you can then add it to your crontab.

The following is an example of how to set up a crontab entry for SampleLoader on Solaris:

1 Log in as the user you want to run SampleLoader.

NOTE It is not necessary to be root to install or run SampleLoader.

2 Display the current jobs:

```
% crontab -l
```

3 To make a new crontab entry, run crontab -e. (Be sure your `EDITOR` environment variable is set to your preferred editor before invoking this command.)

```
% crontab -e
```

4 Enter the appropriate information to run your SampleLoader script. The following example would run sampleloader.sh at 11:30 p.m. every day:

```
30 23 * * * /export/home/SampleLoader/sampleloader.sh >
   /dev/null 2>&1
```

To have SampleLoader output sent to an administrator, use the `cron mailto` command.

For more information on configuring cron entries, enter `man crontab` at the UNIX command line.

Command Line Options

There are six command line options for SampleLoader. To use these options you must append them to the command in the shell script that invokes SampleLoader. For example, to use the `-owner` command line option, you would edit the last line in the script described above to read as follows:

```
$JAVA -classpath $CLASSPATH SampleLoader -config
   SampleLoader.xml -owner moreau
```

The following options are available:

Table 1 Command Line Option

Argument	Usage	Description
-config	-config *xmlConfigFilename*	Provides the filename of the XML configuration file for SampleLoader. This argument is **required**.
-login	-login *username*	Provides a login ID for GeneSpring Workgroup Server, overriding the login ID (if any) specified in the configuration file. This argument is **optiona**l.
-password	-password *password*	Provides a password for GeneSpring Workgroup Server, overriding the password (if any) specified in the configuration file. This argument is **optional**.
-owner	-owner *SampleOwnerName*	Provides a sample owner name for GeneSpring Workgroup Server, overriding the default sample owner name (if any) specified in the configuration file. This argument is **optional**.
-sampleCount	-sampleCount *n*	Directs SampleLoader to load *n* (integer count) samples. If a sampleCount is not specified, SampleLoader loads all samples that meet the criteria defined in the configuration file. This argument is **optional**, and can be used to test connectivity by loading only one or a few samples. Using a count of 0 allows you to test connectivity without loading any data.
-version	-version	Displays the version of SampleLoader you are using (**optional**).

If you run SampleLoader with no arguments, a message appears similar to the following:

```
Thu Dec 19 18:26:03 PST 2002 SEVERE : SampleLoader: command
   line arguments: -config xmlConfigFilename [-login
   loginName -password password] [-owner sampleOwnerName]
   [-sampleCount n] [-version]
```

NOTE

Before running SampleLoader it is a good idea to verify that the XML configuration file is properly formatted. You can do this in most XML editors, or by opening the file in a web browser.

Configuring SampleLoader

This chapter describes the use of the SampleLoader configuration file. You must customize this file for your system before running SampleLoader. You can have any number of different configuration files

Seven example XML files are included in the SampleLoader directory. Pick the one that most closely matches you data loading needs and modify the necessary tags. To edit any of these files, open them in a text editor and change settings as necessary.

The following sections describe the available configuration options and how to use them.

Configuration File Reference

This section contains a list of the tags used in the SampleLoader configuration file (`SampleLoader.xml`). The configuration file is in XML format and uses tags enclosed in angle brackets much like an HTML document.

In such a document, an *element* consists of a tag enclosed in angle brackets, and usually includes a closing tag. For example, the top-level element of this configuration file is the External Database Configuration element. This element consists of opening and closing tags, i.e.:

```
<ExternalDatabaseConfiguration>
...
</ExternalDatabaseConfiguration>
```

An element's *contents* are tags or text nested between the current element's opening and closing tags. The following is an example of elements with contents:

```
<GeNetLoginInfo>
   <UserName>BioMan</UserName>
</GeNetLoginInfo>
```

In the above example, the <UserName> element (including its own contents and its closing tag) is the contents of the <GeNetLoginInfo> element. The username BioMan is the contents of the <UserName> element.

Attributes are values defined within the opening tag of the element itself. In the following example, name is an attribute, and "dbname" is the value of the <PhysicalDatabase> element:

```
<PhysicalDatabase name="dbname">
```

An element may have any number of attributes or contents. An empty element (an element that has attributes but no contents) can be closed within the opening tag by adding a slash at the end, i.e., <tag value="example" />

Tag Reference Table

This table provides a list of all available tags for the SampleLoader configuration file. For more detailed information on the use of each tag, see "Tag Definitions" on page 14.

The Element column lists each of the available tags. The Contents column lists the type of contents that tag can contain (i.e., plain text, or the names of the tags it can contain). The Attributes column lists the attributes that tag can contain.

Table 2 Tag Reference Table for SampleLoader

Element	Contents	Attributes
<ExternalDatabaseConfiguration>	<GeneralConfiguration> <Database>	n/a
<GeneralConfiguration>	<LoadClass> <ProcessedDataListFile> <GeNetLoginInfo>	n/a

Table 2 Tag Reference Table for SampleLoader (continued)

Element (continued)	Contents	Attributes
<Database>	<PhysicalDatabase> <TechnologyType> <Header> <GenomeNames> <GetSampleIDs> <GetSampleAttributes> <GetFile> <GetRawData> <GetSampleProjects>	name, icon
<LoadClass>	plain text	n/a
<ProcessedDataListFile>	plain text	n/a
<GeNetLoginInfo>	<UserName> <Password> <GeNetAddress> <SampleOwner>	n/a
<PhysicalDatabase>	<UserName> <Password> <URL> <Prefetch>	name
<TechnologyType>	n/a	name
<Header>	<Author> <Research_Group> <Organization>	n/a
<GenomeNames>	<GenomeMappingSpec>	n/a
<GenomeMappingSpec>	n/a	targetName sourceName baseDirectory
<UserName>	plain text	n/a
<Password>	plain text	n/a

Table 2 Tag Reference Table for SampleLoader (continued)

Element (continued)	Contents	Attributes
<GeNetAddress>	plain text	n/a
<SampleOwner>	plain text	n/a
<URL>	plain text	n/a
<Prefetch>	plain text	n/a
<Author>	plain text	n/a
<Research_Group>	plain text	n/a
<Organization>	plain text	n/a
<GetSampleIDs>	<DatabaseQuery> <DataDirectory> <FileNameMask> <IDFromFileName> <JavaQuery>	location
<GetSampleAttributes>	<DatabaseQuery> <JavaQuery>	cacheable numeric
<MakeLocation>	n/a	prefix suffix
<GetFile>	<DatabaseQuery> <JavaQuery>	type location deleteAfterwards mimeType
<GetRawData>	<DatabaseQuery> <Format>	n/a
<DatabaseQuery>	SQL command	useGenomeName db
<DataDirectory>	plain text	n/a
<FileNameMask>	plain text	n/a

Table 2 Tag Reference Table for SampleLoader (continued)

Element (continued)	Contents	Attributes
<IDFromFileName>	<RegexpMatch> <DatabaseQuery> plain text	n/a
<RegexpMatch>	plain text	n/a
<JavaQuery>	n/a	class extraArgs
<GetSampleProjects>	<FixedProject> <DatabaseQuery>	n/a
<FixedProject>	plain text	n/a
<Format>	<GeneColumn> <Headlines> <SignalColumn> <NormalizedColumn> <ReferenceColumn> <SignalBackgroundColumn> <ReferenceBackgroundColumn> <ExperimentWorkedColumn> <ExperimentWorkedDesignation> <ExperimentAbsentDesignation> <ExperimentMarginalDesignation> <RegionColumn> <TreatNoSignalAsInvalid> <LowerBoundOnSignalColumn> <UpperBoundOnSignalColumn> <StandardDeviationSignalColumn> <ColumnHeaderLine>	type
<GeneColumn>	plain text	n/a
<Headlines>	plain text	n/a
<SignalColumn>	plain text	n/a
<NormalizedColumn>	plain text	n/a

Table 2 Tag Reference Table for SampleLoader (continued)

Element (continued)	Contents	Attributes
<ReferenceColumn>	plain text	n/a
<SignalBackgroundColumn>	plain text	n/a
<ReferenceBackgroundColumn>	plain text	n/a
<ExperimentWorkedColumn>	plain text	n/a
<ExperimentWorkedDesignation>	plain text	n/a
<ExperimentAbsentDesignation>	plain text	n/a
<ExperimentMarginalDesignation>	plain text	n/a
<RegionColumn>	plain text	n/a
<TreatNoSignalAsInvalid>	plain text	n/a
<LowerBoundOnSignalColumn>	plain text	n/a
<UpperBoundOnSignalColumn>	plain text	n/a
<StandardDeviationSignalColumn>	plain text	n/a
<ColumnHeaderLine>	plain text	n/a

Tag Definitions

<ExternalDatabaseConfiguration>

The top-level element defining the entire SampleLoader configuration. This element contains all of the other tags.

Contents `<GeneralConfiguration>, <Database>`

Attributes n/a

Usage

```
<ExternalDatabaseConfiguration>
...
</ExternalDatabaseConfiguration>
```

Comments Required, can appear only once in the configuration file.

<GeneralConfiguration>

The element containing all of the general configuration options for the Sample Loader.

Contents `<LoadClass>, <ProcessedDataListFile>, <GeNetLoginInfo>`

Attributes n/a

Usage
```
<GeneralConfiguration>...</GeneralConfiguration>
```

Comments Required, can appear only once in the configuration file.

<Database>

The element containing the specifics of the source or sources of sample data, whether it is in a database or a directory of flat files. You must have one Database section for each source to which you will connect. If your data are in flat files, enter a short descriptor of your sample source. The icon attribute is optional, and allows you to specify a graphical image to represent the database.

Contents
```
<PhysicalDatabase>, <TechnologyType>, <Header>,
   <GenomeNames>, <GetSampleIDs>, <GetSampleAttributes>,
   <GetFile>, <GetRawData>, <GetSampleProjects>
```

Attributes *name* (required), *icon*

Usage
```
<Database name="Affymetrix Database"
   icon="/usr/local/graphics/icon.gif">
   ...
</Database>
```

Comments Required.

<LoadClass>

Loads the driver that connects to the database. In some cases you may want to use a JDBC driver written in Java which must be instantiated at startup. You can specify any number of these drivers. Any class you specify, however, must be in your `CLASSPATH`. This element is optional. If you are using a default driver, this is not necessary, but if you are using a specific driver, you must specify it here.

Contents plain text

Attributes n/a

Usage `<LoadClass>sun.jdbc.odbc.JdbcOdbcDriver</LoadClass>`

Comments Optional, frequently used.

<ProcessedDataListFile>

This setting specifies where SampleLoader will save the list of samples that have been uploaded.

Contents plain text

Attributes n/a

Usage
```
<ProcessedDataListFile>/usr/SampleLoader/ProcessedList.txt</
   ProcessedListDataFile>
```

Comments Required.

<GeNetLoginInfo>

This information is used to log in to GeneSpring Workgroup to load the samples. The UserName, Password, and SampleOwner can be provided as a command line argument to override the values specified here. SampleOwner is an optional value, and is used only for loading samples to GeneSpring Workgroup.

Contents `<UserName>, <Password>, <GeNetAddress>, <SampleOwner> (optional)`

Attributes n/a

Usage `<GeNetLoginInfo>...</GeNetLoginInfo>`

Comments Required.

<PhysicalDatabase>

You must have one Physical Database tag for each physical SQL database to which you will connect. The `name` attribute specifies the database name for any `<DatabaseQuery>` tags that occur within the `<Database>` element. If you are loading data from flat files, you may not use the `<PhysicalDatabase>` element.

Contents `<UserName>, <Password>, <URL>, <Prefetch>`

Attributes *name*

Usage `<PhysicalDatabase name="dbname">...</PhysicalDatabase>`

Comments Required to retrieve files from a SQL database.

<TechnologyType>

This sets the special field “technology type” for each sample uploaded. This field identifies the chip or technology used for the sample to the type indicated.

Contents n/a

Attributes *name*

Usage `<TechnologyType name="Affymetrix"/>`

Comments Optional.

<Header>

This element specifies header fields to be set for each sample uploaded from the current database.

Contents `<Author>, <Research_Group>, <Organization>`

Attributes n/a

Usage `<Header>...</Header>`

Comments Required for `<PhysicalDatabase>`.

<GenomeNames>

This setting allows you to associate samples with genomes. One database may have several genomes. Within this element there must be at least one `<GenomeMappingSpec>` tag.

Contents `<GenomeMappingSpec>`

Attributes n/a

Usage `<GenomeNames>...</GenomeNames>`

Comments Required, can appear only once in a configuration file.

<GenomeMappingSpec>

This element specifies the name of the genome in your sample source, the target genome on GeneSpring Workgroup to upload it into, and the base directory to use for `<GetSampleIDs>` or `<GetFile>` elements. For samples loaded from a directory, this line should appear only once. When pulling samples from a database, if the "*useGenomeName*" attribute of the <DatabaseQuery> tag is set to "true", you can use `<GenomeMappingSpec>` multiple times to pull samples from multiple genomes. However, if the "useGenomeName" attribute is set to false in the `<DatabaseQuery>` tag for `<GetSampleIDs>`, this tag must appear only once.

The attribute values for this tag are as follows:

- targetName—The exact name of the genome on GeneSpring Workgroup into which the samples will be loaded
- sourceName—The name of the database or directory from which to retrieve samples
- baseDirectory—The directory to use for `<GetSampleIDs>` or `<GetFile>` elements if no `<DataDirectory>` is specified

Contents n/a

Attributes *<targetName>, <sourceName>, <baseDirectory>*

Usage

```
<GenomeMappingSpec targetName="Yeast" sourceName="YeastDB"
   baseDirectory="/"/>
```

NOTE In the above example the `baseDirectory` attribute value is "`/`". The second slash (`/`) takes the place of the `</GenomeMappingSpec>` tag.

Comments Required.

<UserName>

This element specifies the username for logging into GeneSpring Workgroup or a sample database.

Contents plain text

Attributes n/a

Usage `<UserName>Ndege MacKenzie</UserName>`

Comments Required for both `<GeNetLoginInfo>` and `<PhysicalDatabase>`.

<Password>

This element specifies the password for logging into GeneSpring Workgroup or a sample database.

Contents plain text

Attributes n/a

Usage `<Password>dbPassword</Password>`

Comments Required for both `<GeNetLoginInfo>` and `<PhysicalDatabase>`.

<GeNetAddress>

Specifies the IP and port number of the GeneSpring Workgroup Server to which SampleLoader will connect. This element is contained by `<GeNetLoginInfo>`.

Contents plain text

Attributes n/a

Usage `<GeNetAddress>192.168.10.10:8080</GeNetAddress>`

Comments Required.

<SampleOwner>

Specifies the GeneSpring Workgroup user that will be designated as the owner of the samples being uploaded. It is generally best to make a group the sample owner, since only the samples' owner and the GeneSpring Workgroup administrator can view them. This user or group must already exist in GeneSpring Workgroup and have access to the necessary genomes. The value contained within the tag must exactly match the user or group name on GeneSpring Workgroup. This element is contained by `<GeNetLoginInfo>`.

Contents plain text

Attributes n/a

Usage `<SampleOwner>YeastGroup</SampleOwner>`

Comments Optional. If left blank the sample owner defaults to the GeneSpring Workgroup Server user name designated by the `<UserName>` tag.

<URL>

Specifies the physical address of the database or directory from which you will retrieve sample data.

Contents plain text

Attributes n/a

Usage `<URL>jdbc:odbc:database</URL>`

Comments Required for `<PhysicalDatabase>`.

<Prefetch>

This is an optional, rarely-used element that allows you to specify how many rows to retrieve from the database during the prefetch process. This may be useful in certain cases where there are performance issues. This element is contained by `<PhysicalDatabase>`.

Contents plain text

Attributes n/a

Usage `<Prefetch>20</Prefetch>`

Comments Optional, rarely used.

<Author>

Specifies the sample author to be designated in the header field for each sample being uploaded.

Contents plain text

Attributes n/a

Usage `<Author>Juanita Nguyen</Author>`

Comments Required for `<Header>`.

<ResearchGroup>

Specifies the research group to be designated in the header field for each sample being uploaded.

Contents plain text

Attributes n/a

Usage `<ResearchGroup>Discovery Central</Research Group>`

Comments Required for `<Header>`.

<Organization>

Specifies the organization to be designated in the header field for each sample being uploaded.

Contents plain text

Attributes n/a

Usage `<Organization>Cures R Us</Organization>`

Comments Required for `<Header>`.

<GetSampleIDs>

This element specifies the location from which to upload samples. There are three accepted values for the “location” attribute:

- `database`—perform a database search
- `directory`—locate files in a directory
- `java`—upload files based on the result of a Java call

These values are case-insensitive.

Contents `<DatabaseQuery>, <DataDirectory>, <FileNameMask>, <IDFromFileName>, <JavaQuery>`

Attributes *`location`*

Usage `<GetSampleIDs location="database">...</GetSampleIDs>`

Comments Required if you have a `<PhysicalDatabase>` tag.

<GetSampleAttributes>

Specifies parameters for retrieving attributes associated with samples. These can include the name, value, or units of the sample attribute, and possibly a flag specifying whether the attribute is numeric. The “cacheable” attribute defines whether to cache sample attribute values for previously-uploaded samples. This can greatly improve performance for uploads from external databases. This will not affect automatic upload performance, since in this case sample attributes are already retrieved only once. Acceptable values are “true” and “false”.

The “numeric” attribute indicates whether the retrieved values should be considered numeric. Acceptable values are “yes”, “no”, or “guess”. This attribute is used only for database queries. If the Java query option is used, this setting is overridden.

Contents `<DatabaseQuery>, <JavaQuery>`

Attributes *cacheable, numeric*

Usage
```
<GetSampleAttributes cacheable="true"
   numeric="guess">...</GetSampleAttributes>
```

Comments Optional.

<MakeLocation>

This option specifies the location of data not kept in a database, i.e., a file or a URL. This is done by building a path from a prefix, sample identifier, and suffix. For example:

```
<GetFile type = "Sample Image" location="file"
   mimeType="image/gif">
   <MakeLocation prefix="/fred/chip" suffix="*.gif"/>
</GetFile>
```

If the location is a file, and the file path names obtained here do not start with an absolute path, the baseDirectory attribute of the `<GenomeMappingSpec>` section is prefixed (i.e., `<GenomeMappingSpec baseDirectory="filepathname">`). If the resulting path

name still does not start with an absolute path, it is assumed to be relative place-name to the current user directory (usually the one where program started).

Contents n/a

Attributes *prefix, suffix*

Usage

```
<MakeLocation prefix="directory_path"
   suffix=".file_extension">
```

Comments Optional.

<GetFile>

Specifies parameters for retrieving associated files. This element has four attributes.

The "*type*" attribute specifies the type of file to be retrieved. This attribute is required, and is case-insensitive. There are seven possible values:

- Sample Image—a picture or pictures of the biological sample (e.g. a picture of the individual or the tissue)
- Array Image—a picture or pictures of the scanned array(s)
- CEL File—an Affymetrix CEL file (in GeneSpring Workgroup Server this is actually stored as a general attachment with MIME type `application/x-AffyCELFile`)
- Raw Data File—a raw data file or files
- Signal Raw Data File—a raw signal data file or files (Imagene file format only)
- Control Raw Data File—a raw control data file or files (Imagene file format only)
- attachment—a general attachment (this can be any type of file)

The "*Signal Raw Data File*" and *"Control Raw Data File"* attributes are provided for Imagene users. Use these tags to pair your files, using `<MakeLocation>` once for each file. The `<MakeLocation>` suffix attribute may use the "*" character as a wildcard. For example:

```
<GetFile type="signal raw data file" location="file">
   <MakeLocation prefix="/raw data files/data_"
   suffix="_Cy3*.txt"/>
</GetFile>

<GetFile type="control raw data file" location="file">
   <MakeLocation prefix="/raw data files/data_"
   suffix="_Cy5*.txt"/>
</GetFile>
```

where the files are something like `data_1001_Cy3_xxxx.txt` and `data_1001_Cy5_yyyy.txt`.

The "*location*" attribute specifies the location of the files to be retrieved. This attribute is required, and is case-insensitive. There are four possible values:

- `database`—returns the contents of the file (typically a binary large object, also known as a "blob")
- `file`—returns a file path name
- `URL`—returns a URL
- `java`—returns `com.siGenetics.ext.database.getFile`

The "*deleteAfterwards*" attribute specifies whether to delete the file once it has been uploaded to GeneSpring Workgroup. Accepted values are "true" and "false". This attribute applies only if the "*location*" attribute is set to "file". This attribute is optional. If not specified, its value defaults to "false".

The "*mimeType*" attribute specifies the MIME type of the file or files being retrieved. Any valid MIME type is an acceptable value. This attribute is optional.

Contents `<DatabaseQuery>, <JavaQuery>`

Attributes *`type, location, deleteAfterwards, mimeType`*

Usage

```
<GetFile type="Sample Image" location="database"
   deleteAfterwards="true"
   mimeType="image/gif">...</GetFile>
```

Comments Optional.

<GetRawData>

This option specifies how to retrieve the actual sample data. This may come from either a database or a raw file (or files). The raw file itself may have been a file downloaded from a database or extracted from a Java class.

If the data is located in a database, use `<DatabaseQuery>` to retrieve it. If it is in a file or directory of files, it is interpreted as a tab-delimited file, and you must specify the file format using the `<Format>` tag.

Contents `<DatabaseQuery>, <Format>`

Attributes n/a

Usage `<GetRawData>...</GetRawData>`

Comments Required.

<DatabaseQuery>

This element allows you to enter a SQL query that produces a list of sample identifiers, attributes, or other data based on the provided genome name. If "*useGenomeName"* is true, the SQL query is passed the sourceName specified in the current `<GenomeMappingSpec>` tag. Use the *"db"* attribute to specify the database to query.

Accepted values for "*useGenomeName*" are "true" and "false".

The data retrieved by this option varies depending on which tag contains it, as follows:

- `<GetSampleIDs>`—a list of sample identifiers
- `<GetSampleAttributes>`—three columns (or a multiple of three columns, in which case each set of three is considered independently). These three columns are:
- sample attribute value
- sample attribute name
- sample attribute units

Each row represents one attribute. If there is more than one set of three columns, for each set of three, each row represents an attribute.

- <GetFile>—if the “location” attribute is set to “database”, returns two or three columns:
- the data
- the filename
- the mime type (if present, this overrides the mimeType specified in <GetFile>)

Each row in the result represents a file to be loaded.

Contents SQL command

Attributes *useGenomeName, db*

Usage

```
<DatabaseQuery useGenomeName="true" db="dbname">select ID
   from Experiments where
   Experiments.chipType=?</DatabaseQuery>
```

Comments Required if the location attribute value in <GetSampleIDs> is “database”. For more usage examples, see the sample configuration files included with the SampleLoader distribution.

<DataDirectory>

If samples are contained in flat files rather than a database, this setting specifies the directory in which sample files are located. If a directory is not specified or does not begin with an absolute path, the baseDirectory attribute of the <GenomeMappingSpec> tag is used.

Contents plain text

Attributes n/a

Usage <DataDirectory>/usr/share/affy</DataDirectory>

Comments Optional.

<FileNameMask>

FileNameMask is applied to all files in DataDirectory to filter the FileNames. If the DataDirectory is not specified or does not begin with an absolute path, the baseDirectory attribute of the current <GenomeMappingSpec> section is used. If baseDirectory does not begin with an absolute path, the current user directory is used.

Contents plain text

Attributes n/a

Usage <FileNameMask>*/AffyChipID*.chip</FileNameMask>

Comments Required for retrieving data from flat files in a directory.

<IDFromFileName>

This allows you to generate sample IDs directly from file names. If sample IDs are generated using <Regexpmatch>, only one genome should be specified in the <GenomeNames>/<GenomeMappingSpec> tags. The result of the <RegexpMatch> on the file names provides the sample IDs. If you are using <DatabaseQuery> instead, the file names are passed as arguments to the specified SQL query.

Contents <RegexpMatch>, <DatabaseQuery>, plain text

Attributes n/a

Usage <IDFromFileName>...</IDFromFileName>

Comments Optional.

<RegexpMatch>

When using <IDFromFileName>, use either this tag or <DatabaseQuery>, but not both.

Contents plain text

Attributes n/a

Usage <RegexpMatch>AffyChipID(.*)\.chip</RegexpMatch>

Comments Optional.

<JavaQuery>

Allows you to use a Java class to return an array of identifiers. You specify a command such as:

```
<JavaQuery class="com.pharma.SampleLoader.getParameters"
   extraArgs="Blah"/>
```

and it creates an instance of `com.pharma.SampleLoader.getParameters` using the default constructor. That class should implement `com.agilent.ext.database.GetAttributes`. Then for each attribute, a function is called with arguments of the database identifier, the database genome name, and the extra argument. The return value is an array of `com.agilent.ext.database.Attribute` objects, each with *name, value, units* and *isNumeric* fields.

Contents n/a

Attributes *class, extraArgs (optional)*

Usage

```
<JavaQuery class="com.pharma.SampleLoader.getIDs"
   extraArgs=""/>
```

Comments Optional, applies only to `<GetSampleIDs>`, `<GetSampleAttributes>`, and `<GetFile>`. For additional usage examples, see the sample configuration files included with the SampleLoader distribution.

<GetSampleProjects>

Defines project assigned to a sample, either using a fixed project name, or accessed from a database based on Sample ID.

Contents `<FixedProject>, <DatabaseQuery>`

Attributes n/a

Usage `<GetSampleProjects> ... </GetSampleProjects>`

Comments Optional.

<FixedProject>

Specifies a project name for all samples loaded using the SampleLoader XML file.

Contents plain text

Attributes n/a

Usage `<FixedProject>My Project</FixedProject>`

Comments Optional.

<Format>

Specifies the format of raw data to be retrieved. If this data is in a known format, you can specify it using the “type” attribute. Currently supported types are:

- Incyte Internet Download
- Incyte
- Affymetrix
- Affymetrix Pivot Table
- AtlasImage
- GenePix Results
- Imagene
- ScanArray
- QuantArray
- CodeLink Export
- Amersham CodeLink (Mean)
- Amersham CodeLink (Median)
- Amersham Codelink Expression Report

NOTE Affymetrix Pivot files with more than one sample per file are not supported.

If you are retrieving data from a directory containing data in multiple formats, any files that do not match the format you specify here will be ignored by SampleLoader.

NOTE SampleLoader cannot handle .csv files, but if you are using <JavaQuery> to invoke a program that can read .csv files, they can be imported.

If data are being retrieved from a database as a set of columns using a SQL query, a known format type may not be used and must be explicitly defined using the format described below.

If your data are not in a standard format, you must define the format using the available tags. Columns can be specified either as a number (first column=1) or header. If columns are specified by the header, and data is retrieved from a database using a SQL query, make sure the headers retrieved in the SQL query exactly match the headers specified here. It is a good idea to write your SQL queries as follows:

```
select column1 as "column1" from table_name where ...
```

If a column is not used, you can omit the line or enter -1 (“ for strings).

Contents

```
<GeneColumn>, <Headlines>, <SignalColumn>,
  <NormalizedColumn>, <ReferenceColumn>,
  <SignalBackgroundColumn>, <ReferenceBackgroundColumn>,
  <ExperimentWorkedColumn>, <ExperimentWorkedDesignation>,
  <ExperimentAbsentDesignation>,
  <ExperimentMarginalDesignation>, <RegionColumn>,
  <TreatNoSignalAsInvalid>, <LowerBoundOnSignalColumn>,
  <UpperBoundOnSignalColumn>,
  <StandardDeviationSignalColumn>, <ColumnHeaderLine>
```

Attributes *type*

Usage `<Format type="Affymetrix"/> or <Format>...</Format>`

Comments Required.

<GeneColumn>

Specifies which column in the sample data contains the gene identifier. This tag is used only if your data are in a nonstandard format.

Contents plain text

Attributes n/a

Usage `<GeneColumn>1</GeneColumn>`

Comments Required, if data type is not specified in the `<Format>` tag.

<Headlines>

Number of header lines to skip at the top before further processing. This can usually be determined automatically if the columns are specified by header. This tag does not apply when sample data are retrieved from a database.

Contents plain text

Attributes n/a

Usage `<Headlines>0</Headlines>`

Comments Optional.

<SignalColumn>

Specifies the column containing the raw signal data.

Contents plain text

Attributes n/a

Usage `<SignalColumn>31</SignalColumn>`

Comments Required, if data type is not specified in the `<Format>` tag.

<NormalizedColumn>

Specifies the column containing normalized data.

Contents plain text

Attributes n/a

Usage `<NormalizedColumn>30</NormalizedColumn>`

Comments Optional, rarely used.

<ReferenceColumn>

Specifies the column containing raw reference data. This is typically present in two-color experiments.

Contents plain text

Attributes n/a

Usage `<ReferenceColumn>32</ReferenceColumn>`

Comments Optional.

<SignalBackgroundColumn>

Specifies the column containing the background signal to be subtracted from the main signal before further processing.

Contents plain text

Attributes n/a

Usage `<SignalBackgroundColumn>-1</SignalBackgroundColumn>`

Comments Optional, rarely used.

<ReferenceBackgroundColumn>

Specifies the column containing the background signal to be subtracted from the reference signal before further processing.

Contents plain text

Attributes n/a

Usage `<ReferenceBackgroundColumn>-1</ReferenceBackgroundColumn>`

Comments Optional, rarely used.

<ExperimentWorkedColumn>

Specifies the column containing a flag or flags indicating success of the measurement.

Contents plain text

Attributes n/a

Usage `<ExperimentWorkedColumn>33</ExperimentWorkedColumn>`

Comments Optional, see also `<ExperimentWorkedDesignation>`, `<ExperimentAbsentDesignation>`, `<ExperimentMarginalDesignation>`.

<ExperimentWorkedDesignation>

Specifies the flag in the column specified by `<ExperimentWorkedColumn>` that indicates that the measurement worked well.

Contents plain text

Attributes n/a

Usage `<ExperimentWorkedDesignation>P</ExperimentWorkedDesignation>`

Comments Optional, see `<ExperimentWorkedColumn>`.

<ExperimentAbsentDesignation>

Specifies the flag in the column specified by `<ExperimentWorkedColumn>` that indicates that the measurement did not work well.

Contents plain text

Attributes n/a

Usage `<ExperimentAbsentDesignation>A</ExperimentAbsentDesignation>`

Comments Optional, see `<ExperimentWorkedColumn>`.

<ExperimentMarginalDesignation>

Specifies the flag in the column specified by `<ExperimentWorkedColumn>` that indicates that the measurement worked only marginally.

Contents plain text

Attributes n/a

Usage

```
<ExperimentMarginalDesignation>M</ExperimentMarginalDesignat
   ion>
```

Comments Optional, see `<ExperimentWorkedColumn>`.

<RegionColumn>

Specifies the column that indicates regions to be normalized separately.

Contents plain text

Attributes n/a

Usage `<RegionColumn>-1</RegionColumn>`

Comments Optional, rarely used.

<TreatNoSignalAsInvalid>

Specifies whether a signal of "0" should be treated as blank. If no value is specified for this tag, it defaults to "no". Accepted values are "no" and "yes".

Contents plain text

Attributes n/a

Usage `<TreatNoSignalAsInvalid>no</TreatNoSignalAsInvalid>`

Comments Optional, rarely used.

<LowerBoundOnSignalColumn>

When an error model is known, specifies a lower bound on the signal value.

Contents plain text

Attributes n/a

Usage `<LowerBoundOnSignalColumn>-1</LowerBoundOnSignalColumn>`

Comments Optional, rarely used.

<UpperBoundOnSignalColumn>

When an error model is known, an upper bound on the signal value.

Contents plain text

Attributes n/a

Usage `<UpperBoundOnSignalColumn>-1</UpperBoundOnSignalColumn>`

Comments Optional, rarely used.

<StandardDeviationSignalColumn>

When an error model is known, specifies the standard deviation of the signal value.

Contents plain text

Attributes n/a

Usage

```
<StandardDeviationSignalColumn>-1</StandardDeviationSignalCo
   lumn>
```

Comments Optional, rarely used.

<ColumnHeaderLine>

Specifies the row containing the header names. Usually this can be determined automatically.

Contents plain text

Attributes n/a

Usage

```
<ColumnHeaderLine>-1</ColumnHeaderLine>
```

Comments Optional, rarely used.

www.agilent.com

In this book

The *User Guide* contains procedures for using the GeneSpring Workgroup Server 5.3 SampleLoader.

First edition, October 2006

Agilent Technologies